REMARKS/ARGUMENTS

Claims1-20 are pending in the present application. Claims 1, 2, 4, 5, 7-13, 15-17, and 20 are amended. Reconsideration of the claims is respectfully requested.

Applicants do not concede that the originally filed claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are included only to facilitate expeditious prosecution. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications.

I. 35 U.S.C. § 101

The examiner has rejected claims 1, 7, 8, and 10 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. The applicants have amended claims 1 and 7. In light of the amendments to claims 1 and 7, withdrawal of the rejection is respectfully requested.

II. 35 U.S.C. § 112, Second Paragraph

The examiner has rejected claims 1-10, 11-12, and 15 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. The Applicants have amended claims 1, 2, 4, 5, 7-12, and 15 to correct perceived antecedent basis issues. In light of the amendments to the claims, the present rejection is moot. Withdrawal of the rejection is respectfully requested.

III. 35 U.S.C. § 102, Anticipation

III.A. Claims 1-3, 5-6, 13-16

The examiner has rejected claims and 1-3, 5-6, 13-16 under 35 U.S.C. § 102 as being anticipated by Willen et al., Pub No. 2004/0054999 (hereinafter, *Willen*).

Claim 1, as amended, is as follows:

> 1. A method of queuing threads among processors in a multiple processor system having a plurality of multi-processor modules, wherein each of the plurality of multi-processor modules comprises a plurality of processors, wherein each of the plurality of multi-processor modules is associated with one of a plurality of chip run queues, and wherein each of the plurality of processors is associated with one of a plurality of local run queues, the method comprising the computer implemented systems of:
>
> receiving a first thread to be processed;
>
> identifying the first thread as part of an existing process on a first multi-processor module of the plurality of multi-processor modules;

performing a search for an idle processor, wherein the search is restricted to the plurality of processors of the first multi-processor module associated with the existing process; and

assigning the first thread to either one of the plurality of chip run queues, or one of the plurality of local run queues.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case, each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

Willen does not anticipate claim 1, as amended, because Willen does not teach at least the claim feature of "each of the plurality of multi-processor modules comprises a plurality of processors, wherein each of the plurality of multi-processor modules is associated with one of a plurality of chip run queues, and *wherein each of the plurality of processors is associated with one of a plurality of local run queues*...." Willen discloses driving task dispatching with a virtual switching queue which accepts highest priority tasks from multiple switching queues which in turn are accessed by instruction processors seeking tasks. Overload, plucking and stealing of tasks may be used to effectively balance load within the computer system. Willen focuses closely on processor run queues per processor, and allowing the thread or job to determine which affinity domain to run on. Willen does not disclose providing each of the plurality of processors with its own local run queue.

Since claims 2-3, and 5-6 depend from 1, the same distinctions between Willen and the claimed invention in claim 1 apply for these claims. Therefore, based at least on their dependency from independent claim 1, claims 2-3, and 5-6 are also distinguished.

Claims 13 has been amended to recite features similar to those amended into claim 1. Therefore, by virtue of arguments similar to those presented above, claim 13 is also allowable over Willen. Since claims 14-16 depend from 13, the same distinctions between Willen and the claimed invention in claim 13 apply for these claims. Therefore, the rejection of claims and 1-3, 5-6, 13-16 under 35 U.S.C. § 102 has been overcome.

III.B. Claims 7-12, 17-20

The examiner has rejected claims and 7-12, 17-20 under 35 U.S.C. § 102 as being anticipated by Kimmel et al., Patent No. 6, 105, 053 (hereinafter, *Kimmel*).

Claim 7, as amended, is as follows:

> 7. A method of load balancing threads among processors in a multiple processor system having a plurality of multi-processor modules, wherein each of the plurality of multi-processor modules comprises a plurality of processors, wherein each of the plurality of multi-processor modules is associated with one of a plurality of chip run queues, and wherein each of the plurality of processors is associated with one of a plurality of local run queues, the method comprising the computer implemented steps of:
>
> performing, by an idle processor of the plurality of processors of a first multi-processor module of the plurality of multi-processor modules, a first attempt at a thread steal from a first one of the plurality of local run queues of one of the plurality of processors located on the first multi-processor module for reassignment of a thread to a second one of the plurality of local run queue associated with the idle processor;
>
> responsive to failure of the first attempt, performing a second attempt at a thread steal from one of the plurality of chip run queues associated with a second multi-processor module of the plurality of multi-processor modules; and
>
> assigning the first thread to either one of the plurality of chip run queues, or one of the plurality of local run queues.

Kimmel does not anticipate claim 1, as amended, because Kimmel does not teach at least the claim feature of "each of the plurality of multi-processor modules comprises a plurality of processors, wherein each of the plurality of multi-processor modules is associated with one of a plurality of chip run queues, and *wherein each of the plurality of processors is associated with one of a plurality of local run queues*…." Kimmel discloses utilizing a software abstraction of a non-uniform memory access system hardware representing a hierarchical tree structure to maintain the most efficient level of affinity and to maintain balanced processor and memory loads. The hierarchical tree structure includes leaf nodes representing the job processors, a root node representing at least one system resource shared by all the job processors, and a plurality of intermediate level nodes representing resources shared by different combinations of the job processors. Processors can belong to multiple run queues in the tree hierarchy, and threads can be dispatched at any level in the tree. Rebalance of nodes within the tree proceeds uninhibited all the way up the tree. Kimmel does not disclose providing each of the plurality of processors with its own local run queue.

Since claims 8-12 depend from 7, the same distinctions between Kimmel and the claimed invention in claim 7 apply for these claims. Therefore, based at least on their dependency from independent claim 7, claims 8-12 are also distinguished.

Claim 17 has been amended to recite features similar to those amended into claim 7. Therefore, by virtue of arguments similar to those presented above, claim 17 is also allowable over Kimmel. Since claims 18-20 depend from 17, the same distinctions between Kimmel and the claimed invention in claim 17 apply for these claims. Therefore, the rejection of claims and 7-12, 17-20 under 35 U.S.C. § 102 has been overcome.

IV. 35 U.S.C. § 103, Obviousness

The examiner has rejected claim 4 under 35 U.S.C. § 103 as being unpatentable over Willen et al., Pub No. 2004/0054999 (hereinafter, *Willen*). This rejection is respectfully traversed.

The obviousness rejections are predicated upon the assertions made with respect to Willen. As proved above, the underlying assertions made by the examiner regarding Willen's teachings are incorrect vis-à-vis the independent claims. Specifically, Willen does not teach the feature of, "each of the plurality of multi-processor modules comprises a plurality of processors, wherein each of the plurality of multi-processor modules is associated with one of a plurality of chip run queues, and *wherein each of the plurality of processors is associated with one of a plurality of local run queues*" as recited in the independent claims. For this reason, Willen does not teach all of the features of claim 4, at least by virtue of its dependence on the independent claims. Therefore, the rejection of claim 4 under 35 U.S.C. § 103 has been overcome.

V. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: February 21, 2008

Respectfully submitted,

/Brandon G. Williams/

Brandon G. Williams
Reg. No. 48,844
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants